IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. - 25. (Cancelled)

26. (Original) Method for processing computer programs selectively operable on one or more selected individual processors, comprising:

programming an instruction decoder to accept instruction op codes in excess of a set of instruction op codes required for execution of a program;

providing plural answers from the instruction decoder, including plausible wrong answers; and

selecting a predetermined buffer, thereby permitting further operation with a selected one of the plurality of answers.

27. (Currently amended) The method of claim 26, further comprising: using reconfigurable logic gates for calculating ~~the~~ results of execution of an instruction, the calculation of results of the execution of an instruction including accepting correct data operands and plausible wrong data operands; and outputting correct results along with plausible wrong results.

28. (Original) The method of claim 27, further comprising: using at least a portion of the reconfigurable logic gates for calculating the results of the execution of an instruction; using said portion of the logic gates for accepting correct data operands and plausible wrong data

operands; and using said portion of the logic gates for outputting correct results along with plausible wrong results.

29. (Original) The method of claim 28, further comprising:

providing a key shared with a compiler; encrypting standard op codes with the compiler using the key;

storing the key in more than one memory cell type, including a Read Only Memory (ROM), an Electrically Erasable Programmable Read Only Memory (E.sup.2PROM), and a Random Access Memory (RAM);

expanding key bits in the key into a larger set of bits which control the instruction decoder, signal routing, and logic gate reconfiguration;

using a serial number in ROM in the allocation of logic gates and routing of signals, the serial number communicated to the compiler to inform the compiler of custom allocation and routing; and

using the key for signal routing, and logic gate reconfiguration whether the op codes are encrypted op codes or standard op codes.

30. (Cancelled)

31. (Original) The method of claim 26, further comprising:

providing a key shared with a compiler; encrypting standard op codes with the compiler using the key;

providing correct results and plausible wrong results in an output register; and

coordinating the results in word locations in an output register according to the key.

32. (Original) The method of claim 26, further comprising:

providing program instructions in a pipeline architecture; and

establishing information keys as instruction security commands at a plurality of steps in said pipeline architecture, wherein an arithmetic logic unit (ALU) provides variability of logic circuitry for execution of encrypted op codes or standard op codes that provide standard instruction operation types.

33. (Original) The method of claim 26, further comprising:

providing a key shared with a compiler, the key used, by the compiler to encrypt standard op codes into encrypted op codes; and

using the key to coordinate the variations of the data numeric representations and the encrypted op codes.

34. (Original) The method of claim 33, further comprising using the key to provide a capability of re-allocating memory resources and register resources.

35. (Currently amended) The method of claim [[40]] 34, further comprising:

using a serial number in combination with the key in providing said capability of re-allocating memory resources and register resources; and

using the reallocation of memory and register resources whether the op codes are encrypted or not encrypted.

36. (Original) The method of claim 26, further comprising routing a subset of op codes through an instruction buffer to destination logic gates, which reach a programmable instruction decoder and an instruction interdependency checking logic block.

37. (Currently amended) The method of claim 26, further comprising changing the

programming of the instruction decoder during the execution of a program so that ~~the~~ numeric encodings of input data operands and output data results will change.

38. (Original) The method of claim 26, further comprising:

using reconfigurable logic gates able to calculate results of execution of an instruction; and

reconfiguring the logic gates outputting correct results from the logic gates along with plausible wrong results.

39. (Original) The method of claim 26, further comprising:

providing a key shared with a compiler;

encrypting standard op codes with the compiler using the key;

providing data containing both correct results and plausible wrong results at an output register; and

providing the correct results in word locations in the output register coordinated by the key.

40. (Original) The method of claim 26, further comprising:

using logic for requiring network handshaking; and

further using the network handshaking to provide additional key information for continued operation.

41. (Currently amended) Method of compiling a computer program for use on a selected processor, comprising:

providing instruction op codes in excess of a set of instruction op codes required for execution of a program;

providing instruction op codes to provide plural answers from ~~the instruction~~ a decoder, including plausible wrong answers; and

providing instruction op codes to select a predetermined buffer, thereby selecting one of the plurality of answers.

42. (Original) The method of claim 41, further comprising:

providing a key shared with a compiler; and

encrypting standard op codes with the compiler using the key;

establishing word locations in an output register according to the key.

43. (Original) The method of claim 42, further comprising using a serial number in combination with the key.